REMARKS

Claims 1 - 30 are pending in the application.

Claims 1 - 30 are rejected.

Claims 1 and 14 have been amended.

Claims 41 – 52 have been added. No new matter has been added.

In the Drawings

Figure 4 and the Specification have been amended to address the objections to Figures 4 and 5, respectively. The amendment to Figure 4 includes reference number 400 (as described in, at least, p. 11 of the Specification). The amendment to the Specification address both the objections to Figure 5. The use of "block 528" has been changed to "block 530". No new matter has been added.

Rejections Under 35 U.S.C. §§ 102(b)/102(e)

Claims 1-11, 14-23, and 26 are rejected under 35 U.S.C. §102(b) as being anticipated by Inagami et al., U.S. Patent 4,803,620 ("Inagami"). Claims 1, 12-14, 24, 25, 27, 29, and 30 are rejected under 35 U.S.C. §102(e) as being anticipated by Mohamed, U.S. Patent 6,016,395.

While not conceding that these reference are prior art, but instead to expedite prosecution, Applicants have chosen to respectfully traverse the examiner's rejection. Applicants' arguments are made without prejudice to Applicants' right to establish, for example in a continuing application, the references are not prior art to an invention now or thereafter claimed.

Regarding Inagami, the Examiner cites a EXVP instruction of Figure 2 and a VL and a VST instruction of 4b in support of the 102(b) rejection. In Figures 2 and 4, Inagami illustrates a synchronous communication means in a multi-processor system and scalar and vector object

codes for a program, respectively. (Inagami, col. 5, lines 1-15). In Figure 2, Inagami describes that the EXVP instruction activates the vector processing unit (Inagami, Figure 2). In Figure 4, Inagami describes VL as a vector load and VST as vector store. Inagami Figure 4 further describes a use of VL to load an array in a vector register, and a use of VST to store an array. (Inagami, Figure 4).

Regarding Mohamed, the Examiner cites col. 4, lines 20-34 and Figures 2 and 3 in support of the 102(e) rejection. In the cited sections, Mohamed discloses the operation of a VL.w instruction as loading elements from scalar registers into vector registers (Mohamed, col. 4, lines 20-25).

In contrast, in amended claims 1 and 14, Applicants claim at least one vector data instruction for transferring the vector data directly between a memory and a vector buffer. In the cited sections, neither Inagami nor Mohamed teaches at least one vector data instruction for transferring the vector data directly between a memory and a vector buffer. Accordingly, Applicant respectfully submit that claims 1 and 14 are patentably distinguishable from Inagami and Mohamed. Claims 2-13 depend from claim 1 and are patentably distinguishable from Inagami and Mohamed for at least the same reasons. Claims 15-30 depend from claim 14 and are patentably distinguishable from Inagami and Mohamed for at least the same reasons.

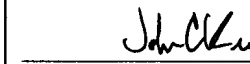
Rejections Under 35 U.S.C. § 103(a)

Claims 14, 27, and 28 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Booth, U.S. Patent 5,247,696 in view of Inagami. However, because claims 27 and 28 depend from claim 14, and because claim 14 has been shown to be patentably distinguishable from Inagami, claim 14, 27, and 28 are patentably distinguishable from Inagami in view of Booth for at least these same reasons.

Accordingly, Applicant respectfully submit that claims 1, 14, and 31 are in condition for allowance. Claims 2-13 depend from claim 1 and are allowable for at least the same reasons. Claims 15-30 depend from claim 14 and are allowable for at least the same reasons. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Applicant has added independent claims 41 and 42 to recite a method and a computer-readable medium having stored thereon instructions which, when executed by a processor, configure the processor to schedule a first transfer of a first stream of vector data between a processor and a memory, and schedule a second transfer of a second stream of vector data between the processor and the memory such that the second transfer is executed in parallel with a computation of data provided by the first transfer. Applicant respectfully submits that claims 41 and 42 are patentably distinguishable from, and thus allowable over, the references cited by the Examiner, the references taken alone or in combination. Claims 43 – 52 depend from claim 42 and are allowable for at least the same reason that claim 42 is allowable.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on November 27, 2002.



Attorney for Applicant(s) — Date of Signature: 11/27/02

Respectfully submitted,

John C. Kennel
Attorney for Applicant(s)
Reg. No. 48,562
Telephone: (512) 439-5080
Facsimile: (512) 439-5099

VERSION WITH MARKINGS TO SHOW CHANGES MADE

The following is a marked-up version of the amended specification paragraphs containing the newly introduced changes, in accordance with 37 C.F.R. § 1.121(b)(1)(iii), showing the changes that the accompanying submission makes to the specification of Application Serial No. 09/375,328. Deleted matter is denoted by bold, bracketed type. Added matter is denoted by bold, underlined type.

In the claims

1. **(Amended)** A method for transferring vector data in a computer system, the method comprising:

identifying use of vector data in an application program;

implementing at least one vector data instruction for transferring the vector data between a memory and a **vector** buffer, the vector data in the buffer being accessible by a processor in the computer system.

14. **(Amended)** A data processing system comprising:

a data processor, **said data processor comprising:**

a cache,

a register file, and

a vector buffer;

means for identifying use of vector data in an application program;

at least one vector data instruction for transferring the vector data **directly** between a memory and **[a] the** buffer, the vector data in the buffer being accessible by the data processor; and

a synchronization instruction to synchronize accessing the vector data with processing the vector data.

41. (New) A method comprising:

a compiler scheduling a transfer of a first stream of vector data between a processor and a memory; and

the compiler scheduling a transfer of a second stream of vector data between the processor and the memory so that the second stream of vector data is transferred while data of the first stream is processed.

42. (New) A computer-readable medium having stored thereon instructions which, when executed by a processor, configure the processor to:

schedule a first transfer of a first stream of vector data between the processor and a memory; and

schedule a second transfer of a second stream of vector data between the processor and the memory such that the second transfer occurs while data of the first stream of vector data is processed by the processor.

43. (New) The computer-readable medium of claim 42 which further configures the processor to:

schedule the first transfer of the first stream of vector data between a vector buffer of the processor and the memory; and

schedule the second transfer of the second stream of vector data between the vector buffer and the memory such that the second transfer occurs while data of the first stream of vector data is processed by the processor.

44. (New) The computer-readable medium of claim 43 wherein at least one of the first transfer and the second transfer is a burst transfer.

45. (New) The computer-readable medium of claim 43 which further configures the processor to:

partition vector data of a computer program into a plurality streams.

46. (New) The computer-readable medium of claim 42 which further configures the processor to:

detect vector data in a computer program.

47. (New) The computer-readable medium of claim 46 which further configures the processor to:

detect a vector data indicator in the computer program, the vector data indicator identifying the vector data.

48. (New) The computer-readable medium of claim 46 which further configures the processor to:

generate a vector instruction, wherein a stream of the plurality of streams is transferred from the memory to the vector buffer in response to execution of the vector instruction.

49. (New) The computer-readable medium of claim 46 which further configures the processor to:

generate a vector instruction, wherein a stream of the plurality of streams is transferred from the vector buffer to the memory in response to the execution of the vector instruction.

50. (New) The computer-readable medium of claim 46 which further configures the processor to:

generate a vector instruction, wherein a stream of the plurality of streams is transferred fom the vector buffer to a register of the processor in response to the execution of the vector instruction.

51. (New) The computer-readable medium of claim 46 which further configures the processor to:

generate a vector instruction, wherein a stream of the plurality of streams is transferred from a register of the processor to the vector buffer in response to the execution of the vector instruction..

52. (New) The computer-readable medium of claim 42 which further configures the processor to:

generate a synchronization instruction to synchronize the first and second transfers while the data of the stream of vector data is processed by the processor.

In the Specification

The paragraph on page 16, lines 1 - 12 has been amended as follows:

There are several exceptions that may be raised with this instruction when an invalid or erroneous operation is attempted. In one embodiment, a first exception that may be raised is the TLB refill exception which indicates that a virtual address referenced by the LDV instruction does not match any of the TLB entries. Another exception is the TLB invalid exception that indicates when the referenced virtual address matches an invalid TLB entry. A third exception that may be raised is the Bus[s] Error exception that indicates when a bus error is requested by the external logic, such as included in memory controller **222**, to indicate events such as bus time out, invalid memory address, or invalid memory access type. A fourth exception is the Address Error exception which indicates that the referenced virtual address is not aligned to a proper boundary.

The paragraph on page 28, lines 12 - 23 has been amended as follows:

In Application programs **132** that handle large amounts of vector data, such as multimedia processing, large block**s** of vector data comprise a major portion of the data used by the program**s**. Performance of D-cache **204** is greatly enhanced with the present invention since VTU **138** offloads D-cache **204** from handling large blocks of vector data. Using VTU **138**, each vector can reside in any page and the cost of switching page boundaries is amortized over the

entire transaction by using long burst transfers. At the application level, the compiler can extract vector streams and exercise an efficient scheduling mechanism to achieve performance improvements. Additionally, scatter/gather operations can be implemented in the present invention by allowing both read and write-back bursts which stride through memory **210**. In contrast, D-cache **204** line fill mechanisms can only implement unit stride transfers efficiently.

The paragraph on page 25, lines 12 - 16 has been amended as follows:

If kernel **506** alternatively performs context switch **522**, second application program **504** resumes execution until finished. Before performing context switch **528**, second application program **504** issues SyncVT and FVB instructions, and bit VBI is cleared, as shown in block **[528]** **530**. Since bit VBI is cleared, bit VBL will be cleared during context switch **524** to first application program **502**.